Efficient Color Image Compression using Block Processing

Alok Kumar[1], Prof. Pankaj Savita [2]
[1]*M.tech. Scholar,* [2]*Guide*
Department of Computer Science & Engineering, School of Research & Technology, People's University, Bhopal (M.P.)

Abstract - With the advancement of internet and communication system multimedia data like image has become most trending object among people. The extensivedemand of the development of transmission and accessing of multimedia data through the telecommunication network and Internet is increased. The Image compression has turned out to be basic for effective transmission and storage of images. With the utilization of digital cameras, requirements for storage, control, and exchange of digital images, has developed violently. These image files can be very large and can possess a great deal of memory. Image data include a critical part of the multimedia data and they involve the significant portion of the communication transfer speed for multimedia data communication. Therefore improvement of productive methods for image compression has turned out to be quite necessary.A fundamental property of image formation for most images is that the neighboring pixels are much correlated and subsequently contain highly redundant information. The basic objective of image compression is to discover a less correlated pixel representation of animage. As recently, the requirement for efficient image compression frameworks can be seen. In the rapidly growing field of Internet applications, not only still images but also small image sequences are used to enhance the design of private and commercial web pages. In this work an adaptive frequency domain block processing for color image compression has proposed and examined based on its PSNR performance in MATLAB image processing environment.

Keywords - Image Processing, Block Processing, Data Compression, Color Image Compression, PSNR, Lossi and Lossless, Compression.

I. INTRODUCTION

One thing that humans always desire is for things to go faster and be more reliable. When think about making online media faster usually direct our thoughts towards being able to view, interact and download files with higher speed. One way of achieving this would be to optimize the way compress images. If the file size of images that interact with gets smaller but keeps the quality then this would mean that they would load faster, giving the user a better experience.

JPEG images use lossy compression. This means that when compress the image will throw away some of the information which usually leads to an overall lower quality and size of the image. The problem is finding the right compression rate. If simplify the JPEG algorithm a little then can say that JPEG uses 4 basic steps: Color conversion, Subsampling, Block-processing with Discrete Cosine Transformation and Variable length encoding. The first step changes the representation of colors in the image. The second step exploits one of the human eye's weaknesses, namely that it is less sensitive to the chrominance information in an image than it is to the luminance. Some of the chrominance information of the image can therefore be removed. During the third step, all the pixels of the image are divided into blocks consisting of 8x8 pixels. These blocks will then be transformed into the frequency-space by the Discrete Cosine Transformation. This will transform the block and separate the low and high frequencies. Because of this, apply quantisation, which will allow removing high frequencies which the human eye cannot see. This is where the most of the actual compression takes place. The last step will reorder the data that is left in the blocks of the image for optimal storage. It is possible for the user to give input to both the second and third step when using an existing JPEG encoder. Depending on the input parameters, the image will be compressed to a certain amount.

The problem is that there is no good way of knowing how much one should compress an image to retain good perceived quality. Using the same compression settings on two different images can yield different results. There might be no visual loss of quality in the first image while the second image can look very poor for the human eye.

In conclusion, the JPEG compression algorithm can be very efficient for certain images but it all depends on the information in the image. There is no "shortcut" to find the best compression setting and obtaining the best possible quality after the compression. By looking at different characteristics of an image it might be possible to determine the best possible image compression, i.e., having the smallest file size but still maintaining an acceptable image quality. This examination work focuses on trying to identifying these characteristics and then uses them to find a close to optimal way of compressing images. An efficient color image compression using block processing approach has proposed and verified based on MATLAB simulation in this work.

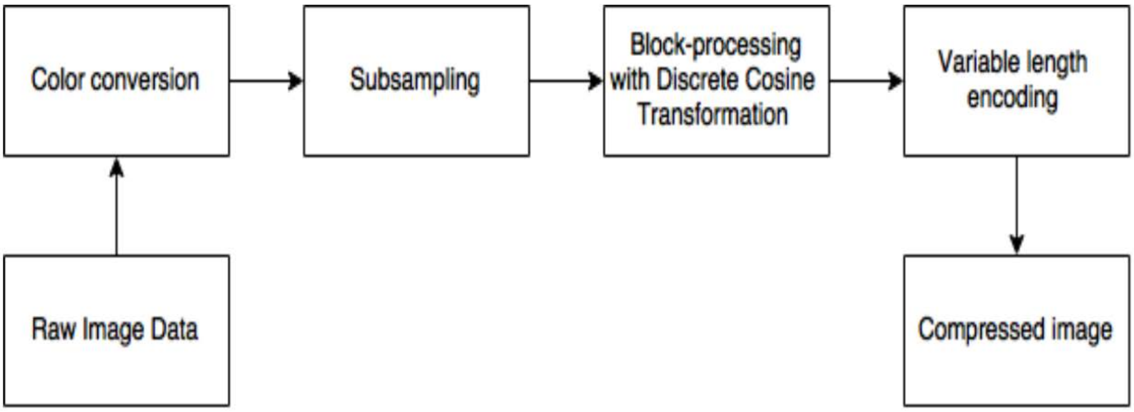

Fig. 1.1 The simplified explanation of JPEG compression

II. SYSTEM MODEL

The need for image compression becomes apparent when number of bits per image is computed resulting from typical sampling rates and quantization methods. For example, the amount of storage required for given images is (i) a low resolution, TV quality, color video image which has 512 x 512 pixels/color,8 bits/pixel, and 3 colors approximately consists of 6 x 10^6 bits;(ii) a 24 x 36 mm negative photograph scanned at 12 x 10^{-6}mm:3000 x 2000 pixels/color, 8 bits/pixel, and 3 colors nearly contains 144 x 10^6 bits; (iii) a 14 x 17 inch radiograph scanned at 70 x 10^{-6}mm: 5000 x 6000 pixels, 12 bits/pixel nearly contains 360 x 10^6 bits. Thus storage of even a few images could cause a problem. As another example of the need for image compression, consider the transmission of low resolution 512 x 512 x 8 bits/pixel x 3- color video image over telephone lines. Using a 96000 bauds (bits/sec) modem, the transmission would take approximately 11 minutes for just a single image, which is unacceptable for most applications.

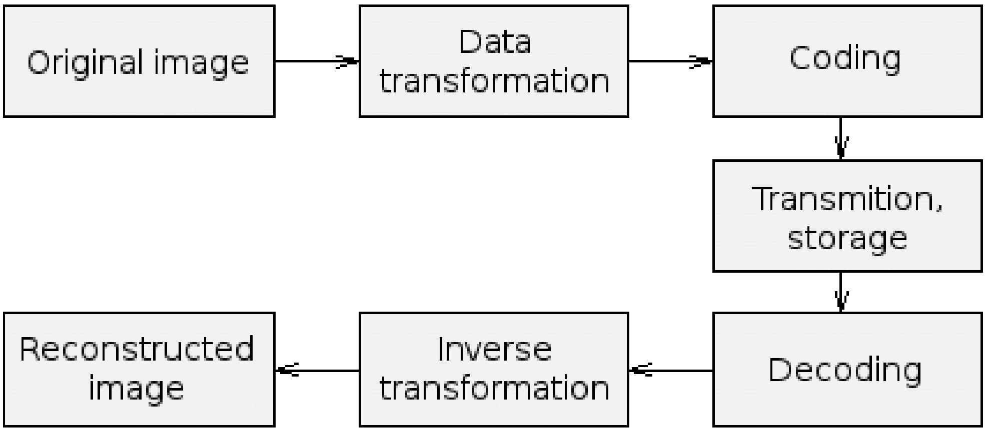

Fig.2.1 conventional Image Compression model

Number of bits required to represent the information in an image can be minimized by removing the redundancy present in it. There are three types of redundancies: (i)spatial redundancy, which is due to the correlation or dependence between neighboring pixel values; (ii) spectral redundancy, which is due to the correlation between different color planes or spectral bands; (iii) temporal redundancy, which is present because of correlation between different frames in images. Image compression research aims to reduce the number of bits required to represent an image by removing the spatial and spectral redundancies as much as possible.

Data redundancy is of central issue in digital image compression. If n1 and n2 denote the number of information carrying units in original and compressed image respectively, then the compression ratio CR can be defined as

$$CR = \frac{n1}{n2}; \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

And relative data redundancy RD of the original image can be defined as RD=1-1/CR;

Three possibilities arise here:

(1) If n1=n2,then CR=1 and hence RD=0 which implies that original image do not contain any redundancy between the pixels.

(2) If n1>>n1,then CR→∞ and hence RD>1 which implies considerable amount of redundancy in the original image.

(3) If n1<<n2,then CR>0 and hence RD→-∞ which indicates that the compressed image contains more data than original image.

III. PROPOSED METHODOLOGY

To overcome the quality and compression ratio an efficient color image compression using block processing approach has been reported in this examination A DCT transform has used for block processing. Transform coding algorithms as a rule begin by dividing the test image into subimages (blocks) of small size (in proposed work 16 × 16). For each block the transform coefficients are determined, viably changing over the original image 16 × 16 array of pixel values into a variety of coefficients inside which the coefficients closer to the upper left corner ordinarily contain a large portion of the data expected to quantize and encode (and inevitably play out the reverse process at the decoder's side) the image with minimal perceptual distortion. The subsequent coefficients are then quantized and the output of the quantizer is utilized by image encoding methods to create the output bitstream representing the encoded image. In image decompression model at the decoder's side, the reverse process takes place, with the conspicuous distinction that the dequantization stage will just create an approximated form of the original coefficient values e.g., whatever loss was presented by the quantizer in the encoder stage is not reversible. Fig. 3.1 shows the block diagram of proposed algorithm. The fundamental blocks of proposed algorithms are as follows:-

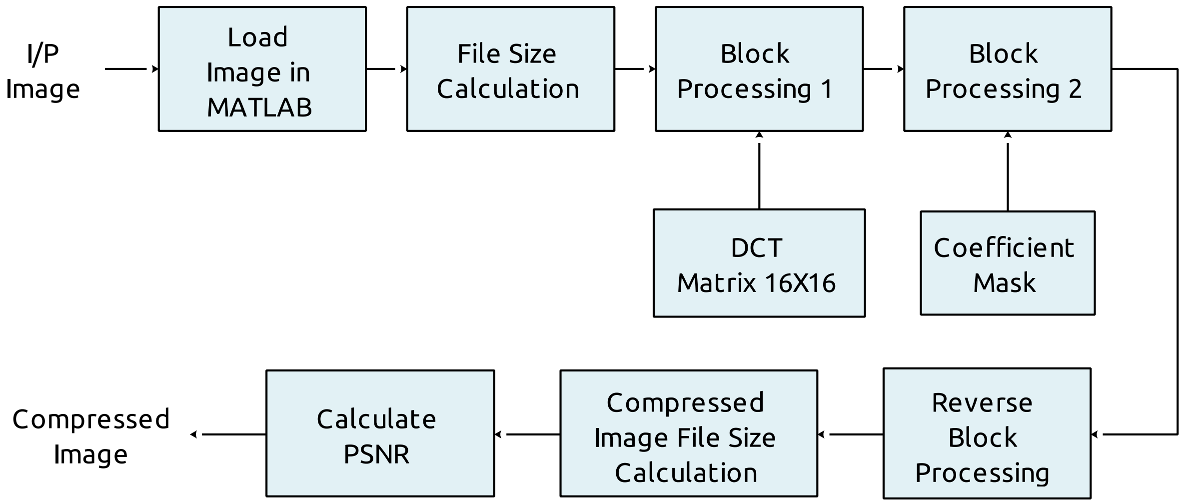

Fig.3.1 Block Diagram of Proposed Methodology.

1. File size calculation

Fir input test image into MATLAB environment and calculate size of original input image before processing it for compression algorithm.

2. Block processing

There are two types of block processing approaches are used to achieve effective compression for big input sample image in block processing 1 DCT transform of 16X16 matrix size block is used in it to process compression. In block pressing 2 coefficient matrices are utilized for processing image compression.

3. Reverse block processing

To reconstruct or retrieve image in its original stage reverse block processing is applied on compressed sample image.

4. Compressed image file size calculation

Now calculate size of file for compresses image after Appling proposed approach. Defiantly the size of retrieved image would be less than uncompressed image but the quality of its visual appearance remains as it is.

5. PSNR calculation

PSNR is a peak signal to noise ratio determined to look at the image quality in the wake of processing through proposed approach. PSNR is a ratio of the peak error. Numerous signals have wide dynamic range, in light of that reason PSNR is generally communicated regarding the logarithmic decibel scale in (dB). Regularly, a higher estimation of PSNR is great since it implies that the proportion of signal to noise is higher. Here, a signal represents to original image and noise represents to the error in reconstruction. It is the proportion between the most extreme conceivable intensity of a signal and the intensity of the corrupting noise. PSNR diminishes as the compression proportion increments for an image. The PSNR is characterized as:

$$PSNR = 10log_{10}\left\{\frac{MAX_1^2}{MSE}\right\}$$
$$= 20log_{10}\left\{\frac{MAX_1}{\sqrt{MSE}}\right\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2)$$

PSNR is figured by estimating the pixel difference between the test image and compressed image.. Fig. 3.2 shows the MATLAB processing flow of proposed examination work.

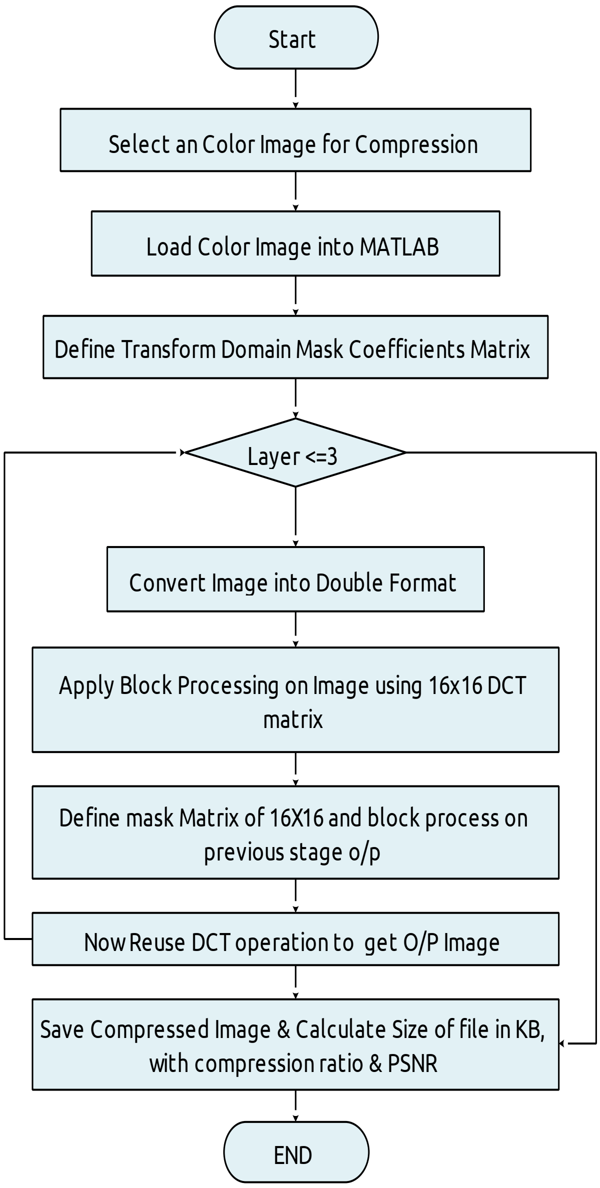

Fig.3.2 Execution Flow Chart of Proposed Methodology

Mean Square Error (MSE)

The MSE is the cumulative squared error between the compressed and the original image. A lower value of MSE means lesser error, and it has the inverse relation with PSNR. Mean square error is a criterion for an estimator: the choice is the one that minimizes the sum of squared errors due to bias and due to variance. In general, it is the average of the square of the difference between the desired response and the actual system output.

$$MSE = \frac{1}{mXn}\sum_{y=1}^{m}\sum_{x=1}^{n}[I(x,y) - \grave{I}(x,y)]^2 \ldots\ldots\ldots\ldots..(3)$$

Where, $I(x,y)$ is the test image (original input image) and $I'(x,y)$ is the reconstructed image (compressed image) and m, n are the dimensions of the image. Lower the value of MSE, the lower the error and better picture quality.

IV. EXPERIMENTAL RESULTS

The performance of the various image compression algorithms is evaluated. The examined algorithms are applied on a few sorts of images: normal images, benchmark images with the end goal that the execution of proposed algorithm can be tested for different applications. These benchmark images are the standard image commonly utilized for the image processing applications. The results of the meticulous simulation for all images and are presented.

In this work prominence quality were given on the measure of compression utilized and how great the reconstructed image be like the original. Investigation was done based on the measure of distortion, which was determined utilizing significant distortion measures: mean square error (MSE), peak signal-to-noise proportion (PSNR) estimated in decibels (dB) and compression proportion (CR) measures were utilized as performance indicators. Image having same PSNR esteem may have distinctive perceptual quality. The nature of reconstructed images can be assessed as far as objective measure and subjective measure. In objective assessment, factual properties are considered while, in subjective assessment, viewers see and research image legitimately to decide the image quality. A decent compression algorithm would recreate the image with low MSE and high PSNR. Execution estimation parameters are depicted in the following sub-sections.

The algorithms were implemented in MATLAB simulation tool. The evaluation parameters (PSNR, MSE), programmed in MATLAB. The proposed algorithm is compared with CSDDS-Based Method. Fig. 4.1 shows the test images taken for simulation and analysis of performance of proposed approach.

Fig. 4.1 Input Uncompressed Images House, Plants, Hat, Yacht, Cablecar, Cornfield, Lena, Airplane, Peppers, Bikes, Coast, Backyard, Boat, Statue and Lighthouse Respectively.

Fig.4.2 Compression Images: Red Layer, Green Layer, Blue Layer and Color Image of Plane Respectively.

Results are tabulated in table 4.1. The results are obtained for images of various images. Original and reconstructed images are also shown in Figure 4.1 and Fig. 4.2. In Fig. 4.2 show the compression images of Red Layer, Green Layer, Blue Layer and Color Image of Plane. It can be seen from table 4.1; the compression ratio CR is high for proposed approach as compare to previous approach based on CSDDS approach. DCT comprises between compression ratio and quality of reconstructed image. Block processing technique is useful in many applications. The graphical representations of compression ratios are as shown in table 4.1 are shown in Fig. 4.3, 4.4, 4.5 and 4.6.

Table 1: Experimental Outcomes of Peak Signal to Noise Ratio (PSNR) for Different Color Image Inputs

Images	***CSDDS-Based Method***				***Proposed (Our)***			
	R	***G***	***B***	***Overall***	***R***	***G***	***B***	***Overall***
House	37.74	35.96	37.62	37.03	56.19	57.20	55.76	***56.38***
Plants	38.18	42.16	36.40	38.31	50.40	49.81	52.88	***51.03***
Hat	47.33	43.81	41.29	43.49	50.19	51.87	51.85	***51.30***
Yacht	41.53	41.96	39.74	40.97	48.33	49.51	48.95	***48.93***
Cablecar	40.73	39.04	39.12	39.56	47.32	48.27	48.05	***47.88***
Cornfield	40.49	39.34	36.37	38.37	46.67	47.36	47.09	***47.04***
Lena	38.16	41.26	38.07	38.93	48.74	49.87	49.84	***49.48***
Airplane	39.78	37.61	40.11	39.02	53.13	53.33	52.58	***53.01***
Peppers	35.20	35.17	34.82	35.06	54.64	56.81	54.44	***55.29***

Bikes	48.58	47.85	46.28	47.46	48.09	48.69	48.58	***48.45***
Coast	49.28	52.20	47.40	49.21	51.34	51.45	51.15	***51.31***
Backyard	43.95	47.71	45.17	45.34	45.72	46.61	46.93	***46.42***
Boat	50.62	53.68	52.20	51.99	44.96	46.24	46.26	***45.82***
Statue	48.08	51.90	47.36	48.35	48.35	48.98	48.73	***48.68***
Lighthouse	49.09	51.91	49.24	49.90	50.09	50.82	50.65	***50.52***
Average	43.25	44.10	42.08	42.89	**49.61**	**50.45**	**50.25**	**50.10**

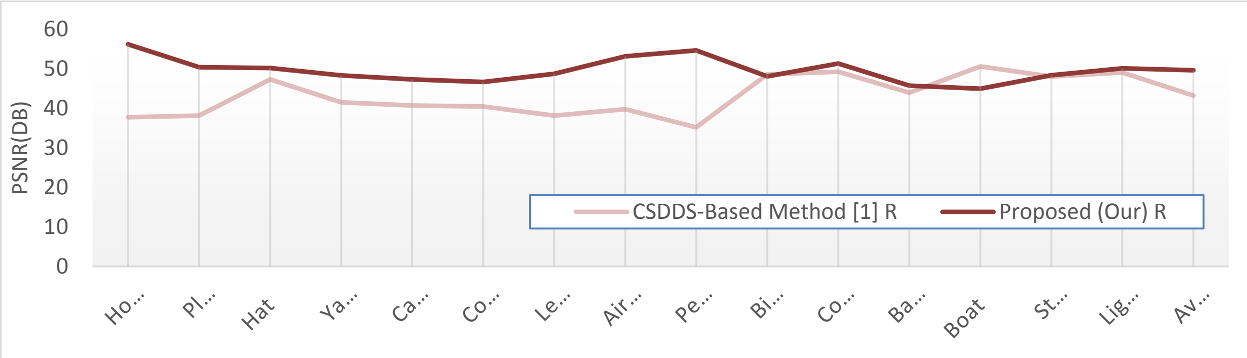

Fig.4.3 Peak Signal to Noise Ratio Comparison of Previous [1] and Proposed (our) Method of RED Layer for House, Plants, Hat, Yacht, Cablecar, Cornfield, Lena, Airplane, Peppers, Bikes, Coast, Backyard, Boat, Statue and Lighthouse Images Respectively.

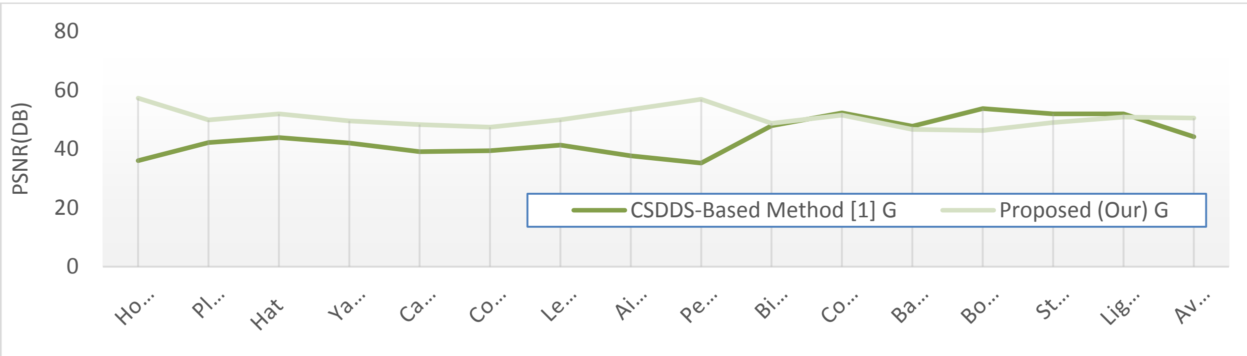

Fig.4.4 Peak Signal to Noise Ratio Comparison of Previous [1] and Proposed (our) Method of GREEN Layer for House, Plants, Hat, Yacht, Cablecar, Cornfield, Lena, Airplane, Peppers, Bikes, Coast, Backyard, Boat, Statue and Lighthouse Images Respectively.

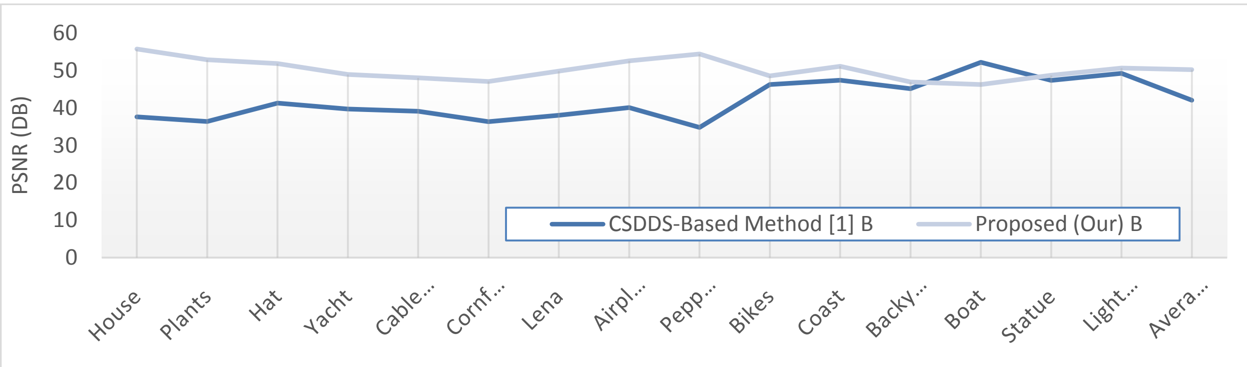

Fig.4.5 Peak Signal to Noise Ratio Comparison of Previous [1] and Proposed (our) Method of BLUE Layer for House, Plants, Hat, Yacht, Cablecar, Cornfield, Lena, Airplane, Peppers, Bikes, Coast, Backyard, Boat, Statue and Lighthouse Images Respectively.

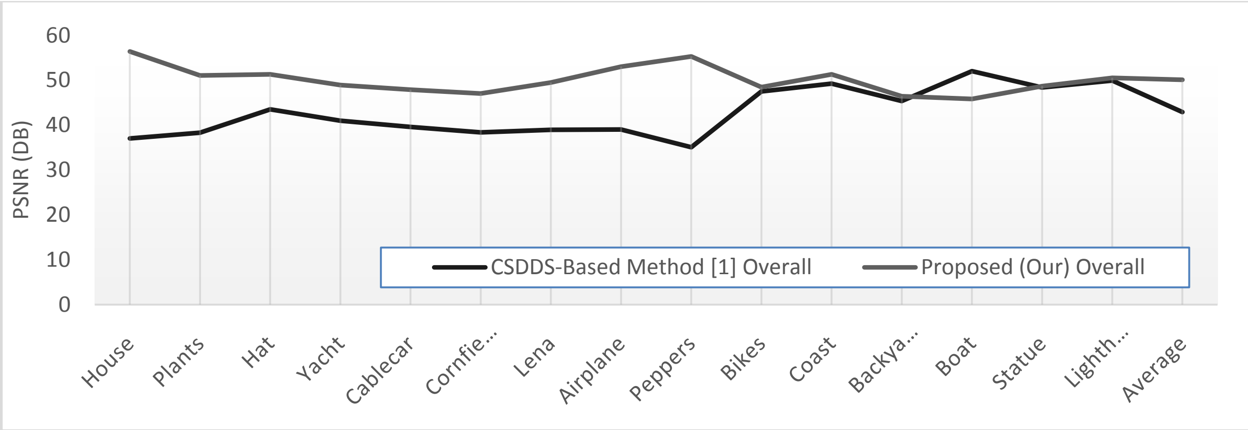

Fig.4.6 Peak Signal to Noise Ratio Comparison of Previous [1] and Proposed (our) Method of Overall RGB Image for House, Plants, Hat, Yacht, Cablecar, Cornfield, Lena, Airplane, Peppers, Bikes, Coast, Backyard, Boat, Statue and Lighthouse Images Respectively.

V. CONCLUSION AND FUTURE SCOPE

In this examination of various Image compression techniques for different images are verified based on parameters, such as mean square error (MSE) and peak signal to noise ratio (PSNR). Proposed simulation results shows that proposed approach can achieve higher compression ratio using proposed technique without affecting quality of image. Standard JPEG compression based on DCT utilizes blocks of image, still there are correlation exits across blocks. This behavior can be explained on the fact that a longer string of continuous zeros can be obtained (after neglecting the similar percentage of pixels) by increasing the block size. This again can be explained on the basis of the fact that an increasing number of symbols are being quantized by the same number of quantization level resulting an increase in quantization error. While for a fixed value of Threshold, compression score/ratio decreases with increase in decomposition Level. Also better compression results are obtained for images of larger size.

The result in this examination work gives a solid establishment to future work for the hardware design. The majority the analysis presented in this examination work included comprehensive simulations in MATLAB. The algorithm can be realized in hardware implementation as a future work.

REFERENCES

[1] S. Zhu, M. Li, C. Chen, S. Liu and B. Zeng, "Cross-Space Distortion Directed Color Image Compression," in IEEE Transactions on Multimedia, vol. 20, no. 3, pp. 525-538, March 2018.

[2] R. R. M. Daga, "Improved k-d tree-segmented block truncation coding for color image compression," 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Singapore, 2017, pp. 178-182.

[3] Q. Zhang, D. Liu and H. Li, "Deep network-based image coding for simultaneous compression and retrieval," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, 2017

[4] C. Jia, X. Zhang, J. Zhang, S. Wang and S. Ma, "Deep convolutional network based image quality enhancement for low bit rate image compression," 2016 Visual Communications and Image Processing (VCIP), Chengdu, 2016, pp. 1-4.

[5] R. Rothe, R. Timofte and L. Van, "Efficient regression priors for reducing image compression artifacts," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 1543-1547.

[6] H. Bai, M. Zhang, M. Liu, A. Wang and Y. Zhao, "Two-Stage Multiview Image Compression Using Interview SIFT Matching," 2014 Data Compression Conference, Snowbird, UT, 2014, pp. 398-398.

[7] Y.-H. Kim, B. C. Choi, and J. Paik, “High-fidelity RGB video coding using adaptive inter-plane weighted prediction,” IEEE Trans. Circuits Syst. Video Technol., vol. 19. no. 7, pp. 1051-1056, Jul. 2009.

[8] X. M. Liu, D. B. Zhao, R. Q. Xiong, S. W. Ma, W. Gao, and H. F. Sun, “Image interpolation via regularized local linear regression,” IEEE Trans. on Image Process., vol. 20, no. 12, pp. 3455-3469, Dec. 2011.

[9] S. Y. Zhu, L. Y. Zeng, B. Zeng, and M. Gabbouj, “Image interpolation based on non-local geometric similarities and directional gradients,” IEEE Trans. on Multimedia, vol. 18, no. 9, pp.1707-1719, Sept. 2016.

[10] H. S. Malvar and D. H. Staelin, “Optimal FIR pre- and postfilters for decimation and interpolation of random signals,” IEEE Trans. on Commun., vol. 36, no. 1, pp. 67-74, Jan. 1988.

[11] W.-S. Lu and A. M. Sevcenco, “Design of optimal decimation and interpolation filters for low bit-rate image coding,” in Proc. IEEE Asia Pacific Conf. on Circuits and Systems, pp. 378-381, 2006.

[12] S. Q. Wang, K. Gu, S. W. Ma, and W. Gao, "Joint chroma downsampling and upsampling for screen content image," IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 9, pp. 1595-1609, Sept. 2016.

[13] T. D. Tran, J. Liang, and C. J. Tu, "Lapped transform via time-domain pre- and post-filtering," IEEE Trans on Signal Process., vol. 51, no. 6, pp. 1557-1571, Jun. 2003.

[14] B. Zeng and J. J. Fu, "Directional discrete cosine transforms: a new framework for image coding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 3, pp. 305-313, Mar. 2008.